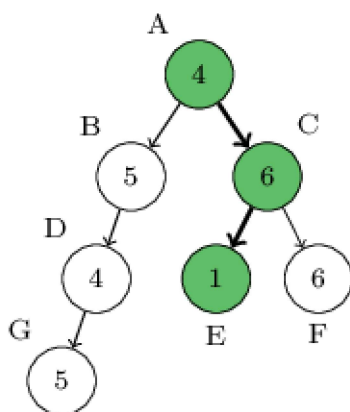


A *binary tree* is either an empty tree or a node (called the *root*) containing a single integer value and linked to two further binary trees. We are interested in paths (sequences of linked adjacent nodes) that start at the root and follow the tree edges (marked as arrows in the figure above). For example, the sequence of nodes A, B, D is a valid path, but the sequence A, B, G is not.

Problem

We would like to find the maximum number of distinct values that appear on a path starting at the root of the tree. For example, on the path consisting of nodes A, B, D, G there are two distinct values (4 and 5). On the path A, C, E there are three distinct values (1, 4 and 6). There is no path that contains four or more distinct values.



Write a function:

```
object Solution { def solution(t: Tree): Int }
```

that, given a binary tree *T* consisting of *N* nodes, returns the maximum number of distinct values that appear on a path starting at the root of tree *T*. For example, given the tree shown above, the function should return 3.

Technical details

A binary tree is given using a pointer data structure. Assume that the following declarations are given:

```
class Tree(var x: Int, var l: Tree, var r: Tree)
```

An empty tree is represented by an empty pointer (denoted by `null`). A non-empty tree is represented by a pointer to an object representing its root. The attribute `x` holds the integer contained in the root, whereas attributes `l` and `r` hold the left and right subtrees of the binary tree, respectively.

Assumptions

Assume that:

- N is an integer within the range $[1..50,000]$;
- the height of tree T (number of edges on the longest path from root to leaf) is within the range $[0..3,500]$;
- each value in tree T is an integer within the range $[1..N]$.

Complexity:

- expected worst-case time complexity is $O(N)$;
- expected worst-case space complexity is $O(N)$.

Copyright 2009–2017 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Link to this task: codility.com/tasks/tree_most_distinct_path/
